APPENDIX I

Appendix I will be understood and appreciated from the following detailed description, taken in conjuction with the drawings in which:

Fig. 72 is a simplified illustration of fast retrieval of data via a windowing query, constructed and operative in accordance with a preferred embodiment of the present invention.

1. Introduction

I.I Scope

Windowing queries play an important role in image processing applications in accordance with preferred embodiments of the present invention. A window is a rectangular region of the image. A windowing query, or a region query, is the task of determining the part of the image contained in a given window, and providing its data. The window concept is a fundamental building block of many tasks in methods according to preferred embodiments of the present invention; windows that define regions of interest for detection, or the defect windows, are created around defects reported by the hardware.

The need to support efficient windowing queries is motivated by the following scenarios:

• Windowing queries during the Inspect scenario:

One method of inspection is based on the comparison of scanned data to reference data (data learned during a learn scan). In many cases however, only a part of the data (around regions of interest) is readily stored for future reference (via top-down windows). During an inspection scan scenario, on defect detection, there is a need to compare the window around the defect with the corresponding region as obtained at learn, even if that window is not a part of a previously stored window. Due to performance constraints, this comparison is preferably done as fast as possible.

• Windowing queries for visualization applications: A similar scenario occurs in applications of data visualization. At the heart of such applications stands the possibility to draw data on the screen at user's request. That is, by marking an area of interest, or a window, the user triggers the application to re-draw the data of that window providing higher level of details. Such interactive applications preferably access their data efficiently and accurately.

1.2 Applicability

The Windowing Reference Manager (WinRef) aims to provide an independent library tool, and to serve as part of the SIP. The WinRef is expected to be applicable to a variety of SIP tasks and data types and structures, and to allow for an efficient data access and retrieval of reference data, e.g., scanned data contained in hardware reports (such as CELs, Color CELs, Features, Snaps, etc.), or in data structures that does not provide an efficient (direct) access to their data items (such as the Chain of Blocks data structure).

2. General Description

2.1 Product Perspective

The Windowing Reference Manager is conceived to be a general purpose, independent library tool, that supports tasks and data types and structures contained in the SIP. Consequently, the target application may impose several restrictions that reduce the tool generality. Preferably the design is as generic as possible, so that its adaptability to other systems will be as simple as possible.

2.2 Product Functions

The fundamental functions of the Windowing Reference Manager needs to support are the insertion of window data items into an internal window data structure, and the efficient retrieval of data items satisfying a given windowing query.

2.3 User Characteristics

The user of the Windowing Reference Manager should be well acquainted with the concepts of the working environment, that is, of C++ and STL. A general knowledge of the SIP system is used for a full understanding of possible applications, and the constraints imposed on the tool design.

3. Functional Requirements

- Purpose - The Windowing Reference Manager (WinRef) is an object class aims to support storage and efficient retrieval operations of window data items.

- Methods The following are the fundamental methods of the WinRef class:

- Construction and Initialization.

- Data Insertion.

- Query operation; Allows for a fast retrieval of data according to some windowing query criterion. The query criterion provides the bounds of the used windowing range.

- Serialization and Deserialization; The capability to store and retrieve information for backup purposes on an external device, such as file, or for transient purposes (e.g. transfer of data between computer systems via TCP/IP). At serialization, sufficient information is stored as to provide a reliable recovery (restore) of the whole window data (deserialization).

4. Interface Requirements The interface to the WinRef class is provided by its methods.

5. General Requirements

5.1 Performance

The tool performance requirements are dictated by those of the target application.

5.2 Error Handling Errors will be handled by return codes.

6. Design Aspects

The following are the design aspects imposed by the expected applications:

In accordance with a preferred embodiment of the present invention, a window may contain two basic types of data: (a) Simple data items such as CELs (Contour Elements), or Features, or, (b) Compound data types such as Polygons (e.g., connected components, or vectors). Simple data items provide local information over a pixel, or sub-pixel, thus, their amount in a single scan is large. Compound data types are fewer in number; nevertheless, each holds more detailed information, and their inter-relations are more complicated. The advantage of using simple data items over the compound ones is that the simple ones are always obtained in order from a scan: first, by lines (the y coordinate), and then by the order within each line (the x coordinate). This observation can be put to use by the future design. Thus, though being a major constraint forced by the target main application, the ordering constraint has the advantage of simplifying the implementation.

Providing that the WinRef tool is designated mainly to support efficient retrieval of simple data items, the order by which items are given can be used to store the items effectively for an efficient future extraction. In specific, if we further assume that data items have a key value according to which the items can be sorted, or ordered, the task of extracting information from windowing reference can be reduced to the task of extracting information from ordered reference.